

УДК 004.89

АВТОМАТИЧЕСКАЯ ДЕТЕКЦИЯ УРОЖАЙНОСТИ ТОМАТОВ НА ОСНОВЕ ТЕХНОЛОГИИ ИИ

Нюпин Олег Максимович, студент 2 курса направления Информационные системы и технологии ФГБОУ ВО РГАУ-МСХА имени К.А. Тимирязева, aureilius@mail.ru

Научный руководитель – Демичев Вадим Владимирович, к.э.н., доцент, доцент кафедры статистики и кибернетики ФГБОУ ВО РГАУ-МСХА имени К.А. Тимирязева, vadi.demiche@mail.ru

Аннотация. Данная статья посвящена обнаружению спелых томатов с помощью искусственного интеллекта. В работе рассмотрены актуальность компьютерного зрения, выделены зависимости, взаимосвязи переменных между собой.

Ключевые слова: анализ данных, искусственный интеллект, сельское хозяйство, компьютерное зрение.

AUTOMATIC TOMATO YIELD DETECTION BASED ON AI TECHNOLOGY

Oleg Maksimovich Nypin, 2st year undergraduate student of the Institute of Economics and Management of the Agro–Industrial Complex, Russian State Agrarian University – Moscow Timiryazev Agricultural Academy, aureilius@mail.ru

Scientific supervisor – Vadim Vladimirovich Demichev, Ph.D. in Economics, Associate Professor, Department of Statistics and Cybernetics, Russian State Agrarian University – Moscow Timiryazev Agricultural Academy vadi.demiche@mail.ru.

Annotation. This article is devoted to the analysis of agricultural vacancy data obtained as a result of parsing. The paper considers the relevance of the visualized data, highlights the dependencies and interrelations of variables among themselves.

Key words: data analysis, artificial intelligence, agriculture, computer vision.

Современный мир характеризуется очень быстрыми темпами развития во всех сферах, включая сельское хозяйство. На текущий момент всюду ведутся разработки интеграции искусственного интеллекта в аграрную сферу. Данная тенденция позволяет предположить увеличение получаемой продукции. Для точного предсказания и прогнозирования урожайности ещё на стадии созревания требуется создать алгоритм, способный распознавать томаты на видео или фотографии.

Требуется создать большой набор данных, в которых будет указано, где находятся томаты. Это позволит тренировать нейросеть на этих данных для последующего предсказания. К сожалению, на данном этапе фотографии приходится размечать вручную, чтобы избежать ошибок со стороны ИИ. Чем более точно размечены данные – тем лучше и быстрее будет работать компьютерное зрение. Для этого поможет сайт Roboflow – инструмент для быстрой и качественной разметки.

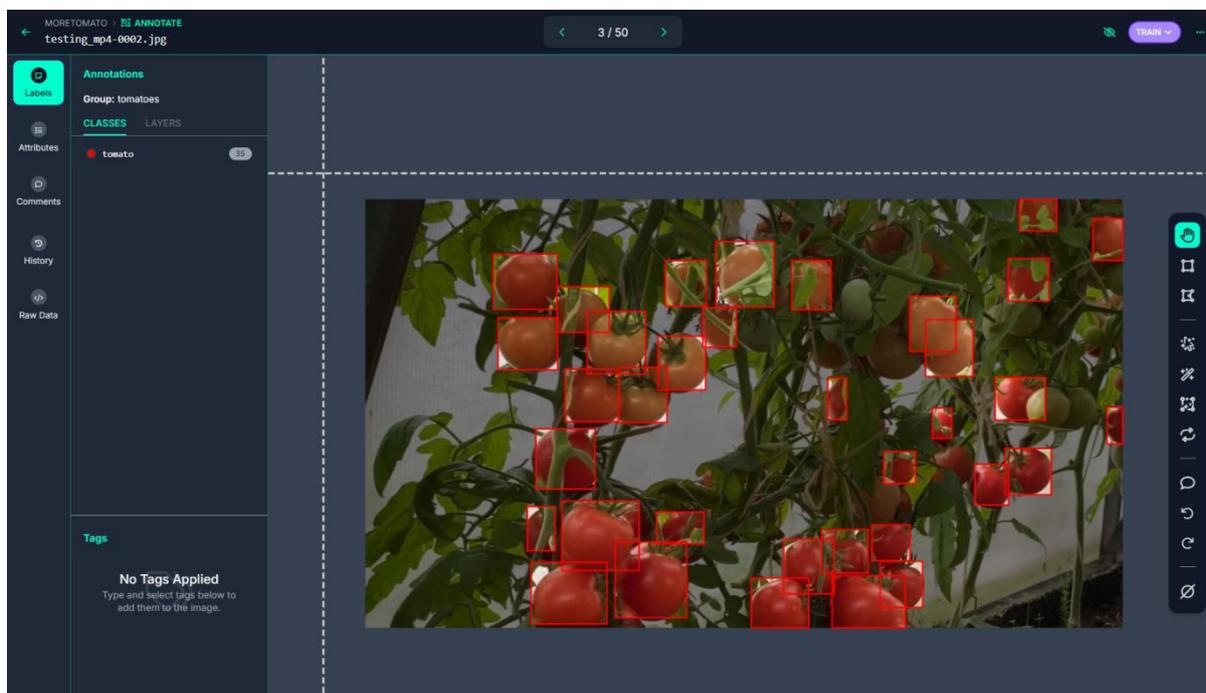


Рисунок 1 – Разметка данных с помощью Roboflow

Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size				
1/100	0G	1.994	2.431	1.51	70	640: 100%	24/24	[06:17<00:00, 15.73s/it]		
	Class	Images	Instances	Box(P	R	mAP50	mAP50-95):	100%	2/2	[00:14<00:00, 7.40s/it]
	all	35	354	0.193	0.799	0.635	0.282			
Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size				
2/100	0G	1.748	1.504	1.364	30	640: 100%	24/24	[05:35<00:00, 13.97s/it]		
	Class	Images	Instances	Box(P	R	mAP50	mAP50-95):	100%	2/2	[00:14<00:00, 7.36s/it]
	all	35	354	0.765	0.489	0.596	0.261			
Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size				
3/100	0G	1.763	1.384	1.349	123	640: 100%	24/24	[05:38<00:00, 14.11s/it]		
	Class	Images	Instances	Box(P	R	mAP50	mAP50-95):	100%	2/2	[00:14<00:00, 7.22s/it]
	all	35	354	0.65	0.656	0.67	0.301			
Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size				
4/100	0G	1.757	1.311	1.341	48	640: 100%	24/24	[05:48<00:00, 14.52s/it]		
	Class	Images	Instances	Box(P	R	mAP50	mAP50-95):	100%	2/2	[00:14<00:00, 7.48s/it]
	all	35	354	0.664	0.508	0.608	0.277			
Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size				
5/100	0G	1.693	1.272	1.325	97	640: 100%	24/24	[05:35<00:00, 13.97s/it]		
	Class	Images	Instances	Box(P	R	mAP50	mAP50-95):	100%	2/2	[00:14<00:00, 7.27s/it]
	all	35	354	0.624	0.59	0.616	0.276			
Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size				
6/100	0G	1.715	1.198	1.312	67	640: 100%	24/24	[05:35<00:00, 13.99s/it]		
	Class	Images	Instances	Box(P	R	mAP50	mAP50-95):	100%	2/2	[00:14<00:00, 7.23s/it]
	all	35	354	0.748	0.74	0.8	0.411			

Рисунок 2 – Тренировка нейронной сети на размеченных данных

Теперь данные должны быть разбиты на тренировочные и тестовые, чтобы отслеживать прогресс обучения нейросети. Данное разбиение позволяет четко понимать процент допущенной ошибки и отклонение от нормы обнаружения. Тренировка происходит с помощью создания модели YOLOv8, которая является на данный момент самым быстрым и точным способом обнаружения объектов на фото или видео. На следующем изображении следует обратить внимание на такие переменные, как `dfl_loss` – регрессия ошибки и `mAP50` – процент правильных результатов.

После того, когда нейросеть научилась определять томаты на фото, можно создать алгоритм, принимающий на вход видеоизображение, которое впоследствии будет обрабатывать. Для этого требуется использовать библиотеки компьютерного зрения, которые могут предоставить широкий функционал обработки изображений. OpenCV – отличный выбор с открытым исходным кодом, который является одним из самых востребованных в сфере компьютерного зрения. С его помощью можно по кадрам разбить видео, затем обработать с помощью ИИ, затем вновь создать видео из полученного результата, где можно будет наблюдать результат работы нашего алгоритма. Также создадим функцию, которая вычисляет вес каждого обнаруженного на видео томата для последующего создания статистики.

```
inference.py > weight_calculation
1 import cv2
2 from ultralytics import YOLO
3 import random
4
5 def weight_calculation(x1,y1, x2, y2):
6     pixel2cm = 0.03
7     pi = 3.14
8     x = abs(x2-x1)
9     y = abs(y2-y1)
10
11     x *= pixel2cm
12     y *= pixel2cm
13     res = int(f'pi * x ** 3 / 5:.0f')
14     return 120 if res > 1000 else res
15
16 def process_video_with_tracking(model, input_video_path, show_video=True, save_video=True, output_video_path='output_video.mp4'):
17     cap = cv2.VideoCapture(input_video_path)
18
19     if not cap.isOpened():
20         raise Exception('Error: Could not open video file.')
21
22     fps = int(cap.get(cv2.CAP_PROP_FPS))
23     frame_width = int(cap.get(cv2.CAP_PROP_FRAME_WIDTH))
24     frame_height = int(cap.get(cv2.CAP_PROP_FRAME_HEIGHT))
25     weights = {}
26
27     if save_video:
28         fourcc = cv2.VideoWriter_fourcc(*'mp4v')
29         out = cv2.VideoWriter(output_video_path, fourcc, fps, (frame_width, frame_height))
30
31     while True:
32         ret, frame = cap.read()
```

Рисунок 3 – Создание алгоритма обработки изображения

Теперь можно использовать заранее подготовленное видео из теплицы для проверки работоспособности алгоритма. Как можно будет увидеть, ИИ отлично справляется с поставленной задачей, четко и предельно точно определяя размеры томатов и вычисляя массу каждого. Данные вычисления сохраняются в виде ключ-значений: номер обнаруженного томата и его средняя масса. Данный метод

позволит избежать повторного подсчета томатов и включения их в сумму найденных несколько раз.

```
inference.py > weight_calculation
1 import cv2
2 from ultralytics import YOLO
3 import random
4
5 def weight_calculation(x1,y1, x2, y2):
6     pixel2cm = 0.03
7     pi = 3.14
8     x = abs(x2-x1)
9     y = abs(y2-y1)
10
11     x *= pixel2cm
12     y *= pixel2cm
13     res = int(f'pi * x ** 3/ 5:.0f')
14     return 120 if res > 1000 else res
15
16 def process_video_with_tracking(model, input_video_path, show_video=True, save_video=True, output_video_path='output_video.mp4'):
17     cap = cv2.VideoCapture(input_video_path)
18
19     if not cap.isOpened():
20         raise Exception('Error: Could not open video file.')
21
22     fps = int(cap.get(cv2.CAP_PROP_FPS))
23     frame_width = int(cap.get(cv2.CAP_PROP_FRAME_WIDTH))
24     frame_height = int(cap.get(cv2.CAP_PROP_FRAME_HEIGHT))
25     weights = {}
26
27     if save_video:
28         fourcc = cv2.VideoWriter_fourcc(*'mp4v')
29         out = cv2.VideoWriter(output_video_path, fourcc, fps, (frame_width, frame_height))
30
31     while True:
32         ret, frame = cap.read()
```

Рисунок 3 – Создание алгоритма обработки изображения

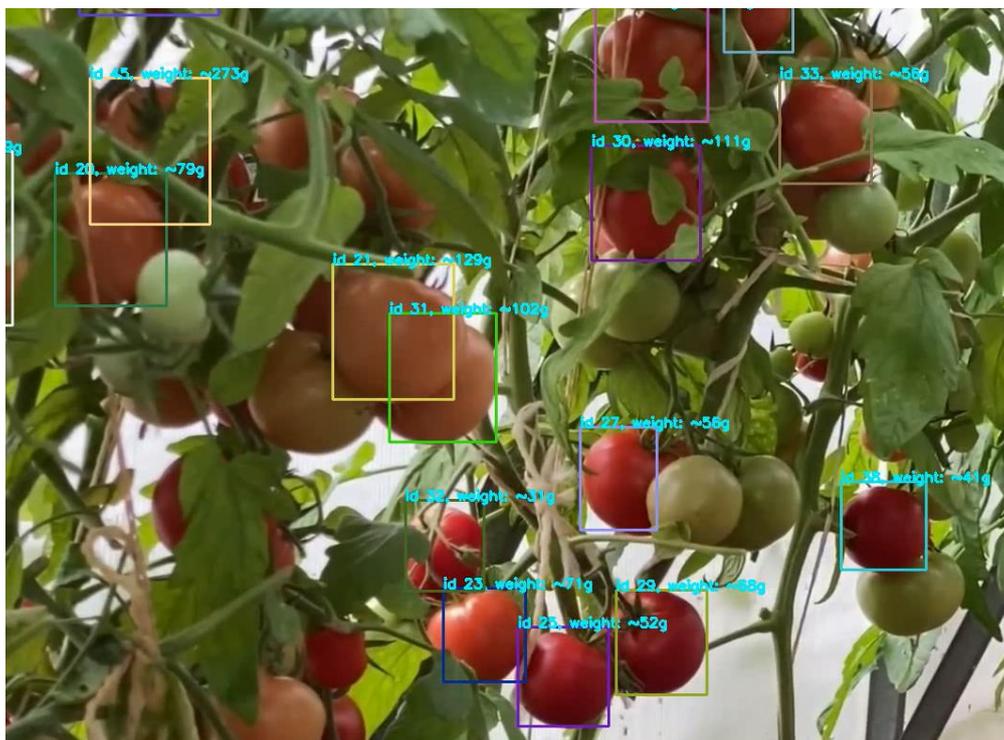


Рисунок 4 – Результат работы программы компьютерного зрения

Конечно, нейросеть ещё обладает неким процентом ошибки, который обуславливается малым количеством тренировочных фото в наборе данных. Для создания особо точной модели требуется не менее 1000 размеченных фотографий. Кроме того, стоит учитывать техническую составляющую модели:

нейросеть использует вычислительные мощности компьютера, и результат напрямую зависит от качества оборудования.

Полученная статистика позволяет оценить количество спелых томатов, а также их средний и общий вес, максимальное и минимальное значения массы. Данный график наглядно показывает среднее распределение массы томатов, что способствует созданию графика и прогнозирования сбора.

Проведенный анализ показал высокую эффективность применения нейронных сетей для распознавания плодов на изображениях и оценки их количества. Использование современных алгоритмов машинного обучения позволяет значительно повысить точность определения урожайности, что имеет важное значение для оптимизации процессов агропромышленного производства.

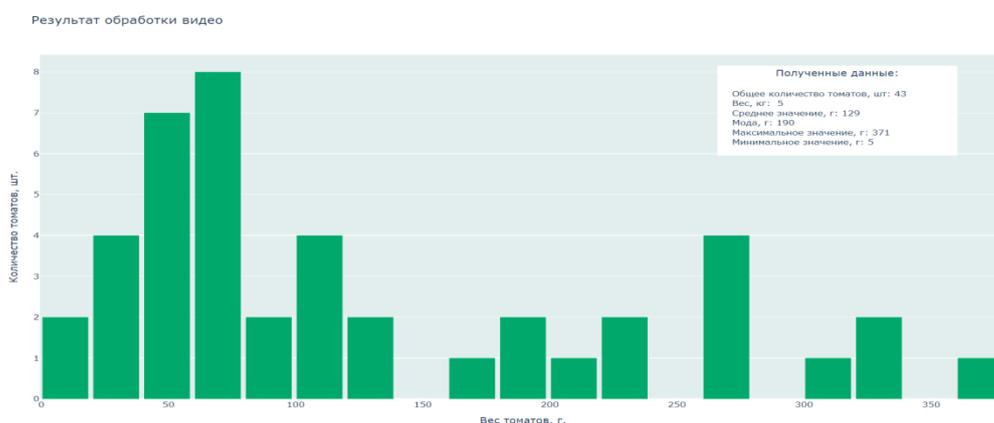


Рисунок 5 – Составленная статистика по обработанному видео

Таким образом, автоматическая детекция урожайности на основе технологии ИИ представляет собой важный шаг вперед в развитии цифровых решений для сельского хозяйства. Дальнейшее совершенствование этих методов позволит еще больше улучшить качество анализа данных и увеличить производительность аграрной отрасли.

Библиографический список

1. Библиотека Pandas [Электронный ресурс] // URL: pandas documentation – pandas 2.1.4 documentation (pydata.org) (дата обращения: 31.10.2024).
2. Библиотека YOLO [Электронный ресурс] // URL: <https://docs.ultralytics.com/> (дата обращения: 31.10.2024).
3. Библиотека Roboflow [Электронный ресурс] // URL: <https://roboflow.com/> (дата обращения: 31.10.2024).
4. Computer Vision: Algorithms and Applications / Richard Szeliski, 2010.
5. Методы анализа данных: Data Mining, Visual Mining, Text Mining / А. А. Барсегян, М.С. Куприянов, В.В. Степаненко, И.И. Холод // OLAP. 2-е изд., перераб. и доп. СПб.: БХВ-Петербург, 2007.
6. Википедия. Свободная библиотека // Интеллектуальный анализ данных в сельском хозяйстве - Data mining in agriculture Интеллектуальный анализ данных в сельском хозяйстве URL:

https://ru.abcdef.wiki/wiki/Data_mining_in_agriculture (дата обращения: 22.12.2023).

7. Невзоров, А. С. Экосистема для анализа больших данных в сельском хозяйстве / А. С. Невзоров, В. В. Демичев // Московский экономический журнал. – 2023. – Т. 8, № 5. – DOI 10.55186/2413046X_2023_8_5_205. – EDN HXDMAK.

8. Титов, А. Д. Использование нейронных сетей для распознавания образов / А. Д. Титов // Сборник трудов приуроченных к 74-й Всероссийской студенческой научно-практической конференции, посвященной 200-летию со дня рождения П.А.Ильенкова, Москва, 01 января – 31 2021 года. – Москва: Российский государственный аграрный университет - МСХА им. К.А. Тимирязева, 2021. – С. 50-53. – EDN SIRZRI.